CERTIFICATE

I, the undersigned, Tatsuo HASEDA, residing at 2-16-16, Saginomiya, Nakano-ku, Tokyo, JAPAN, hereby certify that to the best of my knowledge and belief the following is a true translation into English made by me of Japanese Patent Application Number 2003-041486 filed on February 19, 2003.

Date: December 12, 2006

T. Haseda

Name: Tatsuo Haseda

[Name of Document] Application of Patent

[Reference No.] A000205441

[Application Date] February 19, 2003

[Destination] Commissioner, Patent Office

[International Patent Classification] G06F 15/00

[Title of the Invention] Storage Device, Allotment Range Deciding Method, And Program

[No. of Claims] 20

[Inventor]

    [Address] 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa-ken, R & D Center, TOSHIBA CORPORATION

    [Name] Hideki Yoshida

[Inventor]

    [Address] 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa-ken, R & D Center, TOSHIBA CORPORATION

    [Name] Tatsunori Kanai

[Inventor]

    [Address] 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa-ken, R & D Center, TOSHIBA CORPORATION

    [Name] Nobuo Sakiyama

[Applicant for Patent]

    [Identification No.] 000003078

    [Name] TOSHIBA CORPORATION

[Agent]

    [Identification No.] 100058479

    [Patent Attorney]

    [Name] Takehiko Suzue

    [Telephone No.] 03-3502-3181

[Assigned Agent]

    [Identification No.] 100091351

    [Patent Attorney]

    [Name] Tetsu Kono

[Assigned Agent]

    [Identification No.] 100088683

    [Patent Attorney]

    [Name] Makoto Nakamura

[Assigned Agent]

    [Identification No.] 100108855

    [Patent Attorney]

    [Name] Masatoshi Kurata

[Assigned Agent]

    [Identification No.] 100084618

    [Patent Attorney]

    [Name] Sadao Muramatsu

[Assigned Agent]

    [Identification No.] 100092196

    [Patent Attorney]

    [Name] Yoshiro Hashimoto

[Designation of Charge]

[Ledger No. for Prepayment] 011567

[Amount of Payment] 21000

[List of Filed Document]

[Object Name] Specification 1

[Object Name] Drawings 1

[Object Name] Abstract 1

[Necessity of Confirmation] Necessary

[Name of Document] Specification

[Title of the Invention] Storage device, allotment range deciding method, and program

[What is Claimed is]

[Claim 1]

A storage device used to constitute a distributed storage system, comprising:

means for storing data having an identifier included in an allotment range among an identifier space subjected to by said distributed storage system,

first storing means allotted in said identifier space for storing a standard position as a standard for deciding said allotment range,

second storing means for storing a weight taken into account when deciding said allotment range,

first deciding means, on the basis of said weight stored in said second storing means and a weight relating to a neighboring storage device for allotting a neighboring range of said allotment range in said identifier space, for deciding a relative space width when said identifier space is allotted by said storage device and said neighboring storage device, and

second deciding means, on the basis of said relative space width decided by said first deciding means, for deciding an allotment range within a range from said standard position stored in said first storing means in

4

said identifier space to a standard position relating to said neighboring storage device.

[Claim 2]

A storage device according to Claim 1, wherein said first deciding means adds said weight stored in said second storage means and said weight relating to said neighboring storage device and describes a value obtained by dividing said weight stored in said second storage means by the value obtained by addition as said relative space width.

[Claim 3]

A storage device according to Claim 2, wherein said second deciding means adds a value obtained by multiplying a value indicating said standard position stored in said first storage means by said relative space width and a value obtained by multiplying a value indicating said standard position relating to said neighboring storage device by a value obtained by subtracting said relative space width from 1 and describes a position indicated by a value obtained by addition and a space between said position and said standard position stored in said first storage means as an allotment range within a range between said standard positions.

[Claim 4]

A storage device according to Claim 1, wherein when there are both a first neighboring storage device and a second neighboring storage device as said neighboring

5

storage device, for said first neighboring storage device, said decision by said first deciding means and said decision by said second deciding means are made, thus a first allotment range is decided, and for said second neighboring storage device, said decision by said first deciding means and said decision by said second deciding means are made, thus a second allotment range is decided, and a range including said first allotment range and said second allotment range is described as all allotment ranges to be obtained.

[Claim 5]

A storage device according to Claim 1, wherein when there is only one neighboring storage device as said neighboring storage device, for said neighboring storage device, said decision by said first deciding means and said decision by said second deciding means are made, thus a first allotment range is decided, and a space between said standard position stored in said first storage means and a position, among both ends of said identifier space, of said end close to said standard position stored in said first storage means is decided as a second allotment range, and a range including said first allotment range and said second allotment range is described as all allotment ranges to be obtained.

[Claim 6]

A storage device according to Claim 1 further

comprising a third storing means for storing an address allotted to use for communication with another storage device composing said distributed storage system, wherein: a value obtained by applying a predetermined hash function to said address is described as a value indicating said standard position.

[Claim 7]

A storage device according to Claim 6, wherein a value obtained by applying said predetermined hash function to an address relating to said neighboring storage device is described as a value indicating said standard position relating to said neighboring storage device.

[Claim 8]

A storage device according to Claim 6 or 7 further comprising:

obtaining means for obtaining information indicating addresses allotted to all of or a part of other storage devices composing said distributed storage system and fourth storing means for storing said addresses of said other storage devices obtained by said obtaining means.

[Claim 9]

A storage device according to Claim 8, wherein said addresses relating to said neighboring storage device and said addresses respectively allotted to said other storage devices for allotting identifiers equivalent to values obtained by adding respectively 1, 2, 4, 8, ---, $2^{(b-1)}$ (b

is a predetermined integer) to values obtained by applying a predetermined hash function to said address stored in said third storing means are obtained beforehand by said obtaining means and are stored in said fourth storing means.

[Claim 10]

A storage device according to any one of Claims 6 to 9, wherein said addresses are IP addresses.

[Claim 11]

A storage device according to any one of Claims 1 to 10, wherein one said storage device is brought into correspondence to a plurality of virtual nodes, and different allotment ranges are allotted to the respective virtual nodes, and the allotment ranges of the respective virtual nodes are respectively decided by said first and second means.

[Claim 12]

A storage device according to Claim 11, wherein values obtained by multiplexing identification information of said virtual nodes for said addresses of said storage device and applying said predetermined hash function to it are described as values indicating said standard positions of said virtual nodes.

[Claim 13]

A storage device according to Claim 11 or 12, wherein said respective virtual nodes are weighted.

[Claim 14]

A storage device according to Claim 11 or 12, wherein a common weight is given to all said virtual nodes.

[Claim 15]

A storage device according to any one of Claims 1 to 14, wherein another storage device having a standard position closest to said standard position stored in said first storing means in said identifier space is described as said neighboring storage device.

[Claim 16]

A storage device according to any one of Claims 1 to 14, wherein another storage device having a standard position close to a "n"th position (n indicates a predetermined integer of 2 or larger) for said standard position stored in said first storing means in said identifier space as said neighboring storage device and said decision by said first deciding means and said decision by said second deciding means are made.

[Claim 17]

A storage device according to any one of Claims 1 to 15, wherein said data is files or file blocks.

[Claim 18]

A storage device used to constitute a distributed storage system comprising:

means for storing data having an identifier included in an allotment range allotted by an own device among an identifier space subjected to by said distributed storage

system,

first deciding means, on the basis of a weight given to said own device and a weight given to a neighboring storage device for allotting a neighboring range of said allotment range of said own device in said identifier space, for deciding a relative space width allotted by said own device when said identifier space is allotted by said own device and said neighboring storage device, and

second deciding means, on the basis of said relative space width decided by said first deciding means, for deciding an allotment range allotted by said own device within a range from said standard position allotted beforehand to said own device in said identifier space to said standard position allotted beforehand to said neighboring storage device in said identifier space.

[Claim 19]

An allotment range deciding method of a storage device used to constitute a distributed storage system for storing data having an identifier included in an allotment range among an identifier space subjected to by said distributed storage system comprising:

a step of storing a standard position allotted to said identifier space and used as a standard for deciding said allotment range in a first storing means,

a step of storing a weight considered when deciding said allotment range in a second storage means,

a first deciding step, on the basis of said weight stored in said second storage means and a weight relating to a neighboring storage device for allotting a neighboring range of said allotment range in said identifier space, for deciding a relative space width when allotting said identifier space by said neighboring storage device and said storage device, and

a second deciding step, on the basis of said relative space width decided by said first deciding step, for deciding an allotment range within a range from said standard position stored in said first storing means in said identifier space to a standard position relating to said neighboring storage device.

[Claim 20]

A program for functioning a computer as a storage device used to constitute a distributed storage system for realizing:

a function for storing data having an identifier included in an allotment range among an identifier space subjected to by said distributed storage system,

a first storage function for storing a standard position allotted to said identifier space and used as a standard for deciding said allotment range,

a second storage function for storing a weight considered when deciding said allotment range,

a first deciding function, on the basis of said weight

stored in said second storage function and a weight relating to a neighboring storage device for allotting a neighboring range of said allotment range in said identifier space, for deciding a relative space width when allotting said identifier space by said neighboring storage device and said storage device, and

a second deciding function, on the basis of said relative space width decided by said first deciding function, for deciding an allotment range within a range from said standard position stored in said first storing function in said identifier space to a standard position relating to said neighboring storage device.

[Detailed Description of the Invention]

[0001]

[Background of the Invention]

[Field of the Invention]

The present invention relates to a storage device used for the constitution of a distributed storage system, an allotment range deciding method for deciding an allotment range allotted by the own device among an identifier space subjected to by the distributed storage system, and a program.

[0002]

[Description of the Related Art]

In recent years, a distributed storage system for distributing and arranging many storages in a wide area

network and realizing one virtual storage from demands for grid computing of allotting a process by computers distributed and arranged on the network and correspondence to disasters has been notified.

[0003]

In such a system, the storage nodes composing the system are often increased or decreased due to addition and trouble, so that it is unrealistic to allot manually the file system or files individually to the storages. Further, in a method for installing a server for performing centralized control and intensively managing allotment of the files to the storage nodes, a problem arises that an occurrence of an obstacle in the server and load concentration influence the whole system. Therefore, how to distribute and execute automatic allotment of the files to the storage nodes free of centralized control comes into a problem.

[0004]

To solve this problem, in a researching distributed storage system as described in Non-Patent Document 1 and Non-Patent Document 2, the hash function is applied to the address of a storage node, and the node ID is decided, thus the storage node is mapped in the same space as that of the file ID, and to the storage node having the node ID closest to the file ID possessed by the concerned file, the concerned file is allotted.

[0005]

13

In this method, which storage node a file is to be stored in, if there is a list of node IDs of other storage nodes, can be decided by calculation in one storage node. Therefore, the server for intensively controlling the file allotment is not required and furthermore, there is no need to adjust the file allotment file by file individually between the storage nodes. Therefore, only when increasing or decreasing storage nodes, the addresses of the storage nodes may be notified to other storage nodes, and the communication amount between the storage nodes is decreased, and the processing parallelism is improved.

[0006]

When a storage node is added, at the intermediate point between the added storage node and the neighboring storage node in the hash space, the space is divided again and a part of the space allotted to the existing storage node is inherited and allotted by the new storage node. Inversely, when a storage node is lost, the space allotted to the lost storage node is divided and the divided parts are additionally allotted by both neighboring storage nodes.

[0007]

To prevent files from concentrating upon a specific storage node, the file IDs must be distributed uniformly in the file ID space, so that the file IDs are generally assigned using the hash function. As an argument of the

14

hash function, the file name may be used or the data included in the file may be used.

[0008]

It is possible to divide the files for each block like the CFS and arrange them in the storage nodes for each block or to arrange all the files in the storage nodes like the CAN.

[0009]

[Non-Patent Document 1]

CFS (Wide-area cooperative storage with CFS, Frank Dabek, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica, 18th ACM Symposium on Operating Systems Principles (SOSP ~01), October 2001)

[0010]

[Non-Patent Document 2]

CAN (A Scalable Content-Addressable Network, Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp and Scott Shenker, ACM SIGCOMM 2001)

[0011]

[Problems to be Solved by the Invention]

However, in such a file allotment to the storage nodes by the hash, the expected value of the size of the space allotted to each storage node is uniform in all the storage nodes. Therefore, it cannot respond to the case that the storage capacity, calculation capacity, and line speed are different for each storage node. Therefore, a problem

arises that although in a storage node having a large capacity, the capacity is too much, in a storage node having a small capacity, the capacity is insufficient, and as a whole system, the files cannot be preserved, and also for a storage node having a low calculation capacity, the same amount of I/O as that of a storage node having a high capacity is required, thus the response speed is lowered. When putting the distributed storage system into practical use beyond the research level, this point will be an obstacle.

[0012]

The CFS, to avoid this problem, proposes to make storage nodes having a large capacity virtually correspond to a plurality of storage nodes called virtual nodes. Surely, if a storage node having a large capacity is, for example, several times of a storage node having a small capacity, the large storage node may be divided into several pieces. However, when the capacity is greatly different between the storage nodes, for example, when a storage node having a large capacity has a capacity several thousands times of the capacity of a storage node having a small capacity, the storage node having a large capacity must be divided into several thousands virtual nodes and the management overhead of the virtual nodes comes into a problem. Further, when the mean capacity of the storage nodes is changed due to the progress of the disk technology, how to regulate the unit of virtual nodes also comes into a problem.

Therefore, a single virtual node cannot respond sufficiently to the diversity of capacity.

[0013]

Further, here, the distributed storages in which all the files are arranged in the storage nodes are described mainly, though also in the distributed storages in which the files are divide for each block and are arranged in the storage nodes for each block, a similar problem is caused (similar when considering the case that a set of a file name and a block number is managed as a name of block).

[0014]

The present invention was developed with the foregoing in view and is intended to provide a storage device, when each storage device composing a distributed storage system decides an allotment range among an identifier space subjected to by the distributed storage system, for deciding a more effective allotment range, an allotment range deciding method, and a program.

[0015]

[Means for Solving the Problems]

The present invention is a storage device used to constitute a distributed storage system and includes a means for storing data having an identifier included in the allotment range among the identifier space subjected to by the distributed storage system, a first storing means allotted in the identifier space for storing a standard

17

position which is a standard for deciding the allotment range, a second storing means for storing the weight taken into account when deciding the allotment range, a first deciding means, on the basis of the weight stored in the second storing means and the weight relating to the neighboring storage device for allotting the neighboring range of the allotment range in the identifier space, for deciding a relative space width when the identifier space is allotted by the concerned storage device and neighboring storage device, and a second deciding means, on the basis of the relative space width decided by the first deciding means, for deciding an allotment range within the range from the standard position stored in the first storing means in the identifier space to the standard position relating to the neighboring storage device.

[0016]

Further, the present invention is a storage device used to constitute a distributed storage system and includes a means for storing data having an identifier included in the allotment range allotted by the own device among the identifier space subjected to by the distributed storage system, a first deciding means, on the basis of the weight given to the own device and the weight given to the neighboring storage device for allotting the neighboring range of the allotment range of the own device in the identifier space, for deciding a relative space width

18

allotted by the own device when the identifier space is allotted by the own device and neighboring storage device, and a second deciding means, on the basis of the relative space width decided by the first deciding means, for deciding an allotment range allotted by the own device within the range from the standard position allotted beforehand to the own device in the identifier space to the standard position allotted beforehand to the neighboring storage device in the identifier space.

[0017]

The present invention decides the weight of the storage device on the basis of the storage capacity, calculation capacity, and line speed and performs space allotment according to the weight. For example, to obtain a width in proportion to the weight ratio between the own device and the neighboring storage device, the space between the own node and the neighboring storage device is divided into two parts. By doing this, the space with a width proportional to the weight ratio can be allotted to the storage device.

[0018]

Preferably, it is possible to make one storage device correspond to a plurality of virtual nodes, allot different allotment ranges respectively to the virtual nodes, and decide the allotment ranges of the respective virtual nodes by the first and second deciding means. By allotting a

plurality of virtual nodes to one physical node like this, the allotted space to each storage device can be made more precisely proportional to the weight. When the nodes are assumed, for all the virtual nodes corresponding to one physical node, the probability of neighboring only storage devices having a similar weight is low, so that even if a large width is allotted to one side of the virtual nodes, the total width becomes the size proportional to the weight (it can be avoided that only storage devices with small weights are centralized in the neighboring space and although the capacity is small, a space with a large width is allotted).

[0019]

Further, preferably, it is possible to regard another storage device having a standard position close to the "n"th position (n indicates a predetermined integer of 2 or larger) for the standard position stored in the first storing means in the identifier space as the aforementioned neighboring storage device and make the decision by the first deciding means and the decision by the second deciding means. To realize the reliability which is one of the objects of the distributed storages, there is a case that redundancy of putting one data in a plurality of storage devices is executed. However, by doing this, redundancy in consideration of the weight can be executed. When multiplexing one data by n storage nodes, instead of the

neighboring storage device defined by the standard position, between the storage device ahead n storage devices and the concerned storage device, the space is divided according to the weight and one point is mapped by duplicating on n storage devices. By doing this, even if the storage devices cannot be used partially due to a fault, a disaster, or a line obstacle, by using other storage devices allotting the same space, data can be read or written.

[0020]

Further, the present invention is an allotment range deciding method of a storage device used to constitute a distributed storage system for storing data having an identifier included in an allotment range among an identifier space subjected to by the distributed storage system, comprising a step of storing a standard position allotted to the identifier space and used as a standard for deciding the allotment range in a first storing means, a step of storing the weight considered when deciding the allotment range in a second storage means, a first deciding step, on the basis of the weight stored in the second storage means and the weight relating to the neighboring storage device for allotting the neighboring range of the allotment range in the identifier space, for deciding a relative space width when allotting the identifier space by the neighboring storage device and the concerned storage device, and a second deciding step, on the basis of the relative

21

space width decided by the first deciding step, for deciding an allotment range within the range from the standard position stored in the first storing means in the identifier space to the standard position relating to the neighboring storage device.

[0021]

Further, the present invention is a program for functioning a computer as a storage device used to constitute a distributed storage system and a program for realizing:

a function for storing data having an identifier included in an allotment range among an identifier space subjected to by the distributed storage system,

a first storage function for storing a standard position allotted to the identifier space and used as a standard for deciding the allotment range, a second storage function for storing the weight considered when deciding the allotment range, a first deciding function, on the basis of the weight stored in the second storage function and the weight relating to the neighboring storage device for allotting the neighboring range of the allotment range in the identifier space, for deciding a relative space width when allotting the identifier space by the neighboring storage device and the concerned storage device, and a second deciding function, on the basis of the relative space width decided by the first deciding function, for deciding

22

an allotment range within the range from the standard position stored in the first storing function in the identifier space to the standard position relating to the neighboring storage device.

[0022]

Further, the present invention relating to the device is realized as an invention relating to the method and the present invention relating to the method is realized as an invention relating to the device.

Further, the present invention relating to the device or method is realized also as a program for a computer to execute the procedure equivalent to the concerned invention (or for a computer to function as a means equivalent to the concerned invention or for a computer to realize the function equivalent to the concerned invention) and is realized as a recording medium recording the concerned program which can be read by a computer.

[0023]

According to the present invention, many identifier spaces are allotted to nodes having a large storage capacity and nodes having a fast processing speed by weighting, and more data is allotted to them, thus nodes having a small capacity and nodes having a slow processing speed are prevented from a bottleneck, and the storage capacity and processing speed of the nodes can be used effectively.

[0024]

Further, by combination with assumption, the allotment can be made proportional to the weight more precisely.

[0025]

Further, due to multiplexing and redundancy by duplicated allotment of the space, when any storage node breaks down (the replica of the data is possessed already), the neighboring node inherits immediately the faulty node space and can respond to it, so that high reliability can be realized.

[0026]

Further, according to the present invention, a distributed algorithm for performing space allotment proportional to the weight can be realized without using storage nodes for intensively controlling the whole system, thus the use rate and response speed of the storages can be improved.

[0027]

[Description of the Preferred Embodiments]

Hereinafter, the embodiments of the present invention will be explained with reference to the accompanying drawings.

(First embodiment)

Fig. 1 shows a constitution example of the whole system of this embodiment.

In Fig. 1, numeral 1 indicates a storage node, and 3 indicates a client computer, and 7 indicates a network.

Further, in Fig. 1, three storage nodes composing the distributed storage system are shown, though it is an example, and the number of storage nodes composing the distributed storage system is optional. Further, in Fig. 1, one group of the distributed storage system (storage node group) is shown, though a plurality of groups may exist. In this case, a constitution that each storage node belongs only to any one group and a constitution that there may exist a storage node belonging to a plurality of groups are available.

Further, in Fig. 1, only one client computer is shown, though needless to say, a plurality of client computers may exist.

[0028]

Meanwhile, in the distributed storage system of this embodiment, each storage node has a weight and at time of allotment of the hash space for making the files correspond to the storage nodes, the hash space is divided in a width proportional to each weight between the neighboring storage nodes.

[0029]

Further, in a constitution that both ends exist in the hash space, the two storage nodes allotting the ends of the hash space respectively have one neighboring storage node and the storage nodes other than them respectively have two neighboring storage nodes. Further, in a

constitution that both ends of the hash space are regarded to be connected and the hash space is formed in a loop shape, all the storage nodes have two neighboring storage nodes.

[0030]

As a hash function, it is preferable to use a function having sufficiently long bits for performing uniform allotment. As such a hash function, for example, SHA-1 can be used. The hash space is expressed by an integer of b bits of $0 \sim 2 \wedge b - 1$ (in SHA-1, b = 160).

[0031]

In this embodiment, a plurality of storage nodes 1 distributed and arranged on the network 7 composes one storage node group (distributed storage system). As mentioned previously, a constitution that one storage node 1 belongs to a plurality of storage node groups is available, though in such a case, the hash space may be managed independently for each storage node group.

[0032]

Further, each storage node 1 has an address used for inter-storage node communication, though the node ID (node identifier) of each storage node 1 is a value obtained by applying the hash function to the address possessed by the concerned storage node 1.

Further, hereinafter, as an example of the address, a case that the IP address is used will be explained as an example.

26

[0033]

The storage nodes 1 belonging to a certain storage node group may collect and store not only the addresses possessed by the nodes themselves but also the addresses of all the other storage nodes 1 belonging to the storage node group.

[0034]

However, if each storage node 1 stores the addresses of all other storage nodes 1, for example, when increasing or decreasing the storage nodes 1, the communication amount and processing time may come into a problem, so that it is possible to collect and store, in addition to the address possessed by the own node, only the addresses of a part of the other storage nodes 1 (satisfies the predetermined conditions). For example, each storage node 1 may store the address of the storage node 1 neighboring the own node in the hash space and only the addresses of the nodes allotting the hash space including some points (for example, a plurality of points obtained by adding respectively 1, 2, 4, 8, ---, $2^{(b-1)}$ to the node ID (a value obtained by applying the hash function to the address of the own node) of the own node) in the hash space. By using these addresses, when O (b) nodes are inquired, a storage node 1 for allotting an optional point in the hash space can be searched.

[0035]

Fig. 2 shows an internal constitution example of the

storage node 1 of this embodiment.

As shown in Fig. 2, the storage node 1 has a space width decision unit 11, a space allotment control unit 12, a space allotment information storage unit 13, a file input/output unit 14, and a file storage unit 15. Schematically, the space width decision unit 11, space allotment control unit 12, and space allotment information storage unit 13 decide and manage space allotment and the file input/output unit 14 inputs or outputs files on the basis of the space allotment. Further, the file storage unit 15 stores the file corresponding to the hash space allotted by the own node. Further, the storage node 1 has a storage unit (not drawn) for storing the information concerning the own node (for example, the address of the own node, node ID, weight, etc.).

[0036]

Fig. 3 shows an example of the processing procedure when the storage node relating to this embodiment participates.

The storage node 1, when newly participating in a certain storage node group, firstly, obtains the IP address of the own node (Step S1). As an IP address obtaining method, the obtaining method in the ordinary IP network can be used. For example, a manager may select properly one IP address unused on the sub-network thereof and manually input it to the concerned storage node (hereinafter, referred to

28

as a new storage node) 1 participated. Or, it is also possible to automatically select one from the pooled IP addresses by a DHCP server and notify it to the concerned new storage node 1 by a DHCP protocol.

[0037]

Further, as mentioned previously, the concerned new storage node 1 applies the hash function to the inputted or notified address of the own node, thereby obtains the node ID of the own node (Step S2).

[0038]

Next, the new storage node 1 obtains the IP address of an optional other (for example, one) storage node of the node group and notifies it to the space allotment control unit 12. To obtain the addresses of other storage nodes, any method may be used. For example, the manager properly selects other (for example, one) storage node and may input manually the IP address to the concerned new storage node 1 or when there is a storage node in the neighborhood, by a DHCP option or broadcasting, the IP address of the concerned storage node can be notified automatically from the concerned neighboring storage node to the concerned new storage node.

[0039]

Using the IP address of the other storage nodes obtained in this way, the space allotment control unit 12 of the new storage node 1 is properly connected to other storage

29

node 1 (the space allotment control unit 12 of which),
inquires and collects the addresses of all of or a part
of other storage nodes 1 belonging to the storage node group
to which the own node belongs (Step S3). When storing only
the addresses of a part of the storage nodes 1, for example,
on the basis of the node ID of the own node, the address
of each storage node 1 allotting the neighboring node
address and the hash space including the points with 1,
2, 4, 8, ---, $2^{(b-1)}$ respectively added to the node ID of
the own node are obtained.

[0040]

The addresses of other storage nodes 1 collected in
this way are stored in the space allotment information
storage unit 13 (Step S4).

[0041]

Hereinafter, the hash space dividing method of each
storage node of this embodiment will be explained by
referring to Fig. 4.

The following process, when there are two neighboring
nodes, is performed for each neighboring node.

[0042]

Further, when performing firstly the following process,
for example, it may be performed after Step 4 shown in Fig.
3 or may be performed at the point of time when an I/O request
is firstly received from the client computer or may be
performed at appropriate timing other than it. Further,

30

the following process, for example, when the weight of the own node is changed or when the weight of the neighboring node is changed or when the neighboring node is changed due to an increase or a decrease in the storage nodes, may be performed anew.

[0043]

The space width decision unit 11 of the storage nodes 1 of this embodiment, on the basis of the address of the neighboring node obtained from the space allotment control unit 12, is connected to the neighboring storage node 1 (the space width decision unit 11 of which) and notifies the weight of the own node to it, and obtains the weight of the neighboring storage node 1.

[0044]

As a weight, for example, the storage capacity of the storages, calculation capacity, line speed, and an approximate combination thereof can be used. Here, as an example, a case that the storage capacity of the storages is used will be explained.

[0045]

Assuming the weight of the own node s as V [s] and the weight of the neighboring node u as V [u], the relative width of the own node s is expressed as indicated below.
w = V [s] / (V [s] + V [u])

[0046]

The space width decision unit 11, when deciding space

31

width information w in this way, sends it to the space allotment control unit 12.

The space allotment control unit 11, on the basis of the space width information w, divides the space between the own node s and the neighboring node u. Further, the space between the own node s and the neighboring node u is decided beforehand, and one end point is a point obtained by applying the hash function to the address of the own node u and the other end point is a point obtained by applying the hash function to the address of the neighboring node s.

[0047]

And, for example, assuming the hash function as h(), the address of the own node s as A [s], and the address of the neighboring node u as A [u], when the boundary with the neighboring node u is expressed as follows:

h1 = h (A [u]) *w + h (A [s]) *(1 - w)

the boundary h1 satisfying the following condition is obtained:

h1 - h (A [u]) : h (A [s]) - h1 = V [u] : V [s]

This situation is shown in Fig. 4.

[0048]

The boundary h2 with the other neighboring node d can be obtained by the similar procedure. Namely, assuming the address of the other neighboring node d as A [d] and the weight thereof as V [d], the relative width is expressed

as follows:

w' = V [s] / (V [s] + V [d])

    and when the boundary of the neighboring node d is expresses as follows:

h2 = h (A [d]) *w' + h (A [s]) *(1 - w')

    the boundary h2 satisfying the following condition is obtained:

h2 - h (A [d]) : h (A [s]) - h2 = V [d] : V [s]

    [0049]

    The boundary between each neighboring nodes u and d is decided in this way and the range thereof (refer to h1 and h2 shown in Fig. 4) is stored in the space allotment storage unit 13 as an allotment space of the own node s.

    [0050]

    Further, in a constitution that both ends exist in the hash space, with respect to the two storage nodes allotting the ends of the hash space, the boundary with the neighboring storage node is obtained by the same method as the aforementioned and the interval from the boundary to the end of the hash space on the own node side may be described as an allotment space of the own node.

    [0051]

    The hash space is divided between the neighboring nodes according the weight (for example, the storage capacity) like this (for example, is divided by the width proportional to the weight), thus as illustrated in Fig. 5, for example,

to a storage node having a large storage capacity, more files can be allotted.

[0052]

Further, in the aforementioned, w = V [s] / (V [s] + V [u]) is obtained by the space width decision unit 11 and then h1 = h (A [u]) *w + h (A [s]) *(1 - w) is obtained by the space allotment control unit 12. However, it is possible to obtain h1 = h (A [u]) *V [s] / (V [s] + V [u]) + h (A [s]) *(1 - V [s] / (V [s] + V [u])).

[0053]

Further, the aforementioned deciding method by the space width decision unit 11 and space allotment control unit 12 is an example and various other methods are available.

[0054]

Fig. 6 shows an example of the processing procedure when a request of the storage node relating to this embodiment is received.

The client computer 1, when reading and writing the files in the distributed storage system, sends an I/O request to the storage node 1 of the node group for managing the files used by itself.

[0055]

Further, although an optional storage node 1 of the node group can be used, for practical use, for example, it is supposed to use the storage node 1 at the close position

34

on the network.

[0056]

When the storage node 1 receives the I/O request (Step
S11), the file input/output unit 14 thereof compares the
allotment range of the own node of the space allotment
information storage unit 13 with the file ID (file
identifier) in the request (further, for the file ID, a
value obtained by applying the hash function to the file
name of the objective file or the data of the objective
file is used) (Step S12), and if the file ID is within the
allotment range (Step S13), accesses the file storage unit
15 of the own node, and performs the request process (reading
or writing in the file) (Step S14).

[0057]

On the other hand, when it is beyond the range (Step
S13), on the basis of the address of another node of the
space allotment information storage unit 13, the file
input/output unit 14 inquires another node (the space
allotment control 12 of which), thereby searches for the
address of the storage node 1 for allotting the file ID
thereof (Step S15).

[0058]

And, if the address of the storage node 1 for allotting
the file ID is obtained by searching (Step S16), the file
input/output unit 14 is connected to the storage node 1
(the file input/output unit 14 of which) and performs the

35

request process (Step S17).

[0059]

If the address of the storage node 1 for allotting
the file ID is not obtained (Step S16), the file input/output
unit 14 performs an error process (for example, returns
an error message to the client computer of the request source
(Step S18).

[0060]

As mentioned above, according to this embodiment, by
deciding the space width according to the weight, for
example, file allotment according the storage capacity of
the storage node and load distribution according to the
processing speed can be realized.

[0061]

(Second embodiment)

The distributed storage system of this embodiment
assumes the storage nodes composing one node group (all
or a part of them) as a plurality of virtual nodes, allots
them to a plurality of hash spaces, and then makes them
correspond to the files.

[0062]

Hereinafter, the differences from the first embodiment
will be explained mainly.

[0063]

Fig. 7 shows the hash space dividing method of this
embodiment.

[0064]

When calculating a hash value from the node ID of each storage node (hereinafter, to distinguish from the virtual node, called the physical node) 1, in addition to the address of the physical node, the virtual node number allotted for each physical node is used as an argument of the hash function.

[0065]

For example, assuming the address of a certain physical node s as A [s] and the number of virtual nodes set in the physical nodes s as v, to the physical nodes s, v hash values of h (A [s], 0), ---, h (A [s], v-1) correspond.

[0066]

And, if the virtual nodes are regarded respectively as the storage nodes of the first embodiment, by the same process as that of the first embodiment, the allotment range of the hash space of each virtual node can be obtained.

[0067]

Further, a method for giving a weight to each virtual node and a method for giving a common weight to all the virtual nodes are available.

[0068]

In this embodiment, in addition to the advantage of the first embodiment, the physical node is divided into a plurality of spaces, thereby can neighbor with more nodes, so that the mean value of the weights of the neighboring

nodes approaches the mean value of the weights of all the nodes, thus an advantage is obtained that the space width allotted by each node is proportional to the weight more accurately. Unlike the effect, obtained when assumption is applied to the prior art executing no weighting, that the distribution of distance in the hash space between the nodes is reduced, this is an effect obtained by very combination of space division according to the weight and assumption.

[0069]

(Third embodiment)

The distributed storage system relating to this embodiment allots the space duplicately to a plurality of nodes and allots the file duplicately to the nodes.

[0070]

Hereinafter, the differences from the first embodiment will be explained mainly.

[0071]

Fig. 8 shows the hash space dividing method of this embodiment.

[0072]

Further, Fig. 8 illustrates the case of n = 2 (duplication).

[0073]

When allotting the same point in the hash space duplicately to n nodes, it is possible to divide the storage

nodes into n groups, aim at the same hash space, and in each group and similarly to the first embodiment, obtain the allotment range of the hash space of each storage node.

[0074]

At that time, for example, in place of the neighboring nodes of the first embodiment, between the n storage nodes of the own node in the hash space, the allotment space may be divided. Other storage nodes held between them divide the similar space, so that as a result, one point is duplicately allotted to n nodes.

[0075]

On the other hand, in the case of simple multiplexing, there are various methods for making the files correspond to a plurality of nodes available. For example, a redundancy method for applying a plurality of hash functions to the files, thereby allotting a plurality of virtual file IDs, and storing the files in a plurality of nodes having the node IDs corresponding to those virtual file IDs may be considered. However, if a storage node is lost due to a fault, it is the storage node neighboring with the faulty node in the hash space to inherit the space allotted by the storage node. In such a method, to the storage node neighboring with the faulty node. the concerned file must be transferred from another storage node.

[0076]

On the other hand, in this embodiment, duplicate allotment is performed for the neighboring storage node, so that the storage node inherits the space, thus the transfer amount of files can be minimized.

[0077]

According to this embodiment, in addition to the advantage of the first embodiment, there is an advantage that the files are stored in a plurality of nodes, thus the possibility that files are lost due to trouble is reduced, and at time of trouble, the file transfer amount is minimized, thus the allotment space can be inherited.

[0078]

Further, the third embodiment and fourth embodiment can be combined and executed.

[0079]

Further, in the aforementioned, when storing the files in the storage nodes, the example that each file is stored is explained. However, a case that the files are divided for each block and are stored for each block is available similarly. In this case, for example, a set of the file name and block No. is described as a block name, thus the same method as the aforementioned can be applied.

[0080]

Further, the functions aforementioned can be realized as software.

Further, this embodiment can be executed as a program

40

for a computer to execute a predetermined means (or for a computer to function as a predetermined means or for a computer to realize a predetermined function) and can be executed as a recording medium recording the program which can be read by the computer.

[0081]

Further, the constitutions illustrated by the embodiments of the present invention are examples, and it is not intended to eliminate constitutions other than them, and other constitutions obtained by replacing a part of any illustrated constitution with another one, omitting a part of any illustrated constitution, adding another function or element to any illustrated constitution, or combining them are available. Further, another constitution theoretically equivalent to any illustrated constitution, another constitution including a part theoretically equivalent to any illustrated constitution, and another constitution theoretically equivalent to the essential section of any illustrated constitution are also available. Further, another constitution for accomplishing the same or similar object to any illustrated constitution and another constitution for producing the same or similar effect to any illustrated constitution are also available.

Further, various variations of the various components illustrated in the embodiments of the present invention

41

can be combined and executed properly.

Further, the embodiments of the invention include inventions relating to various viewpoints, stages, concepts, and categories such as an invention as an individual device, an invention of two or more related devices, an invention as a whole system, an invention of components in an individual device, and an invention of a method corresponding to them.

Therefore, from the disclosed contents of the embodiments of the present invention, inventions can be extracted free of limitation to the illustrated constitutions.

[0082]

The present invention is not limited to the embodiments aforementioned and can be modified and executed variously within the technical scope thereof.

[0083]

[Effects of the Invention]

According to the present invention, each storage device composing the distributed storage system, when deciding an allotment range allotted by itself among an identifier space subjected to by the distributed storage system, can decide a more effective allotment range.

[Brief Description of the Drawings]

Fig. 1 is a drawing showing the whole constitution example of the distributed storage system relating to an

42

embodiment of the present invention.

Fig. 2 is a drawing showing the internal constitution example of the storage node relating to the same embodiment.

Fig. 3 is a flow chart showing an example of the processing procedure when the storage node relating to the same embodiment participates.

Fig. 4 is a drawing for explaining the hash space dividing method of the same embodiment.

Fig. 5 is a drawing for explaining the hash space dividing method of the same embodiment.

Fig. 6 is a flow chart showing an example of the processing procedure when a request of the storage node relating to the same embodiment is received.

Fig. 7 is a drawing for explaining another hash space dividing method of the same embodiment.

Fig. 8 is a drawing for explaining still another hash space dividing method of the same embodiment.

[Description of Numerals]

1 … Storage node, 3 … Client computer, 7 … Network, 11 … Space width decision unit, 12 … Space allotment control unit, 13 … Space allotment information storage unit, 14 … File input/output unit, 15 … File storage unit

[Name of Document] Abstract

[Abstract]

[Problem] To provide a storage device, when each storage device composing distributed storages decides an allotment range in a file identifier space subjected to by the distributed storages, for deciding a more effective allotment range.

[Solving Means] A file storage unit 15 of the storage device stores a file having a file identifier included in the allotment range allotted by an own node among the file identifier space subjected to by a distributed storage system. A space width decision unit 11, from a weight possessed by the own node and a weight possessed by a neighboring node, decides a relative space width. A space allotment control unit 12 applies a predetermined hash function to an address of the own node, obtains a first standard position in the file identifier space, similarly obtains a second standard position from an address of the neighboring node, and on the basis of the relative space width, decides an allotment range within a range from the first standard position to the second standard position.

[Selected Drawing] Fig. 1

[Name of Document] Drawing

Fig. 1

7 Network

A The file is read or written via the near node.

1 Storage node

B The pointer to another node is possessed.

3 Client computer

Fig. 2

11 Space width decision unit

A Space width information

B Capacity information

C Space width decision unit for other nodes

12 Space allotment control unit

D Space allotment information

E Space allotment information

F Space allotment control unit for other nodes

13 Space allotment information storage unit

G Space allotment information

14 File I/O unit

H Application

I File I/O request

J File data

K File data

L File I/O unit for other nodes

15 File storage unit

Fig. 3

M Start

S1 Obtain address of own node

S2 Calculate node ID of own node

S3 Collect addresses of other nodes

S4 Store collected addresses of other nodes

N End

Fig. 4

1 Hash space

2 u: Neighboring node

3 Capacity V [u]

4 h1: Upper limit of allotment space of own node

5 Allotment space of own node

6 s: Own node

7 Capacity V [s]

8 h2: Lower limit of allotment space of own node

9 d: Neighboring node

10 Capacity V [d]

Fig. 5

1 Hash space

2 The space between the nodes is divided by the width
proportional to the capacity.

3 File

4 To nodes having a large capacity, many files are allotted.

Fig. 6

A Start

S11 Receive I/O request

S12 Compare allotment range of own node with file ID requested

S13 Is allotment range of own node within allotment range?

S14 Perform request process to file storage unit of own node

S15 Search for address of storage node for allotting file ID requested

S16 Is address of concerned storage node obtained?

S18 Perform error process

S17 Connect to storage node and perform request process

B End

Fig. 7

1 Allot a plurality of spaces to one node (Assumption of node)

Fig. 8

1 Allot duplicately spaces for multiplexing (For duplication, space division proportional to the capacity between two neighboring nodes)

2 File

3 The files are stored in a plurality of nodes.

[FIG. 1]

NETWORK 7

THE FILE IS READ OR
WRITTEN VIA THE NEAR
NODE

1 STORAGE NODE

THE POINTER TO ANOTHER
NODE IS POSSESSED

3 CLIENT COMPUTER

[FIG. 2]

```
         11 ～  ┌──────────────────┐   CAPACITY
              │  SPACE WIDTH     │   INFORMATION      SPACE WIDTH DECISION
              │  DECISION UNIT   │ ◄──────────────►   UNIT FOR OTHER NODES
              └──────────────────┘
                       │  SPACE WIDTH INFORMATION
                       ▼
         12 ～  ┌──────────────────┐   SPACE ALLOTMENT INFORMATION
              │  SPACE ALLOTMENT │ ◄──────────────►   SPACE ALLOTMENT CONTROL
              │  CONTROL UNIT    │                    UNIT FOR OTHER NODES
              └──────────────────┘
                       │  SPACE ALLOTMENT INFORMATION
                       ▼
         13 ～  ┌──────────────────┐
              │  SPACE ALLOTMENT │
              │  INFORMATION     │
              │  STORAGE UNIT    │
              └──────────────────┘
              14 ～   │  SPACE ALLOTMENT INFORMATION
                      ▼
APPLICATION   ┌──────────────────┐                    FILE I/O UNIT
─────────────►│  FILE I/O UNIT   │ ◄──────────────►   FOR OTHER NODES
  FILE T/O    └──────────────────┘   FILE DATA
  REQUEST            ▲  FILE DATA
                     │
               ┌───────────┐
         15 ～ │  FILE     │
              │  STORAGE  │
              │  UNIT     │
               └───────────┘
```

[FIG. 3]

```
                    ┌───────────┐
                    │   START   │
                    └───────────┘
                          │
                          ▼
        ┌──────────────────────────────────────┐
        │   OBTAIN ADDRESS OF OWN NODE          │～ S1
        └──────────────────────────────────────┘
                          │
                          ▼
        ┌──────────────────────────────────────┐
        │   CALCULATE NODE ID OF OWN NODE       │～ S2
        └──────────────────────────────────────┘
                          │
                          ▼
        ┌──────────────────────────────────────┐
        │   COLLECT ADDRESSES OF OTHER NODES    │～ S3
        └──────────────────────────────────────┘
                          │
                          ▼
        ┌──────────────────────────────────────┐
        │ STORE COLLECTED ADDRESSES OF OTHER NODES │～ S4
        └──────────────────────────────────────┘
                          │
                          ▼
                    ┌───────────┐
                    │    END    │
                    └───────────┘
```

[FIG. 4]

HASH SPACE

h(A[u]) ← 1   u : NEIGHBORING NODE

CAPACITY V [u]

V[u])

-- h1 : UPPER LIMIT OF ALLOTMENT
         SPACE OF OWN NODE

ALLOTMENT SPACE OF OWN NODE

V[s])

1

h(A[s]) ← s : OWN NODE

CAPACITY V [s]

V[s])

-- h2 : UPPER LIMIT OF ALLOTMENT
         SPACE OF OWN NODE

V[d])

1

h(A[d]) ← d : OWN NODE

CAPACITY V [d]

[FIG. 5]

HASH SPACE

THE SPACE BETWEEN THE NODES IS DEVIDED BY THE WIDTH PROPORTIONAL TO THE CAPACITY

~1

FILE

TO NODES HAVING ALARGE CAPACITY, MANY FILES ARE ALLOTTED

~1

~1

[FIG. 6]

```
                    ┌─────────┐
                    │  START  │
                    └────┬────┘
                         │
                         ▼
         ┌──────────────────────────────┐
         │  RECEIVE I/O REQIEST          │──── S11
         └───────────────┬──────────────┘
                         │
                         ▼
         ┌──────────────────────────────┐
         │  COMPARE ALLOTMENT RANGE      │──── S12
         │  OF OWN NODE WITH FILE id     │
         │  REQUESTED                    │
         └───────────────┬──────────────┘
                         │
                         ▼
                                    S13
                  ◇─────────────────◇
                 / IS ALLOT-          \        NO
                ◇  MENT RANGE OF OWN    ◇──────────────┐
                 \ NODE WITHIN ALLOT-  /               │
                  \ MENT RANGE ?      /                │
                   ◇────────┬────────◇                 │
     S14                    │ YES                      │ S15
                            ▼                          ▼
   ┌──────────────────────────────┐    ┌──────────────────────────────┐
   │ PERFORM REQUEST PROCESS       │    │ SEARCH FOR ADDRESS OF         │
   │ TO FILE STORAGE UNIT OF       │    │ STORAGE NODE FOR ALLOTT-      │
   │ OWN NODE                      │    │ ING FILE ID REQUESTED         │
   └──────────────┬───────────────┘    └───────────────┬──────────────┘
                  │                                     │
                  │                                     ▼       S16
                  │              NO            ◇──────────────────◇
                  │      ┌─────────────────────◇ IS ADDRESS        ◇
                  │      │                      ◇ OF CONCERNED STOR- ◇
               S18 │     ▼                      ◇ AGE NODE OBTAINED ? ◇
                  │  ┌──────────────┐           ◇─────────┬────────◇
                  │  │ PERFORM ERROR│                     │ YES
                  │  │ PROCESS      │                     │    S17
                  │  └──────┬───────┘                     ▼
                  │         │           ┌──────────────────────────────┐
                  │         │           │ CONNECT TO STORAGE NODE       │
                  │         │           │ AND PERFORM REQUEST           │
                  │         │           │ PROCESS                       │
                  │         │           └───────────────┬──────────────┘
                  │         │                           │
                  │◄────────┴───────────────────────────┘
                  ▼
             ┌─────────┐
             │   END   │
             └─────────┘
```

[FIG. 7]

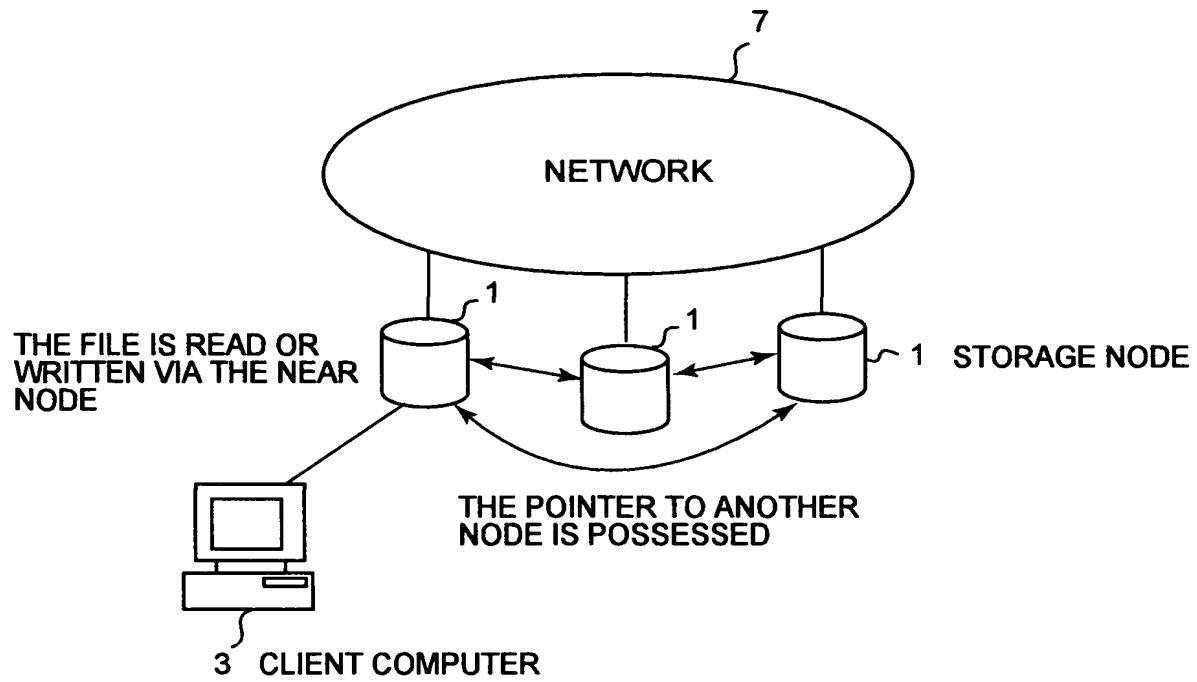ALLOT A PLURALITY OF SPACES TO
ONE NODE (ASSUMPTION OF NODE)

[FIG. 8]

ALLOT DUPLICATELY SPACES FOR MULTIPLEXING
(FOR DUPLICATION, SPACE DIVISION PROPORTIONAL TO THE
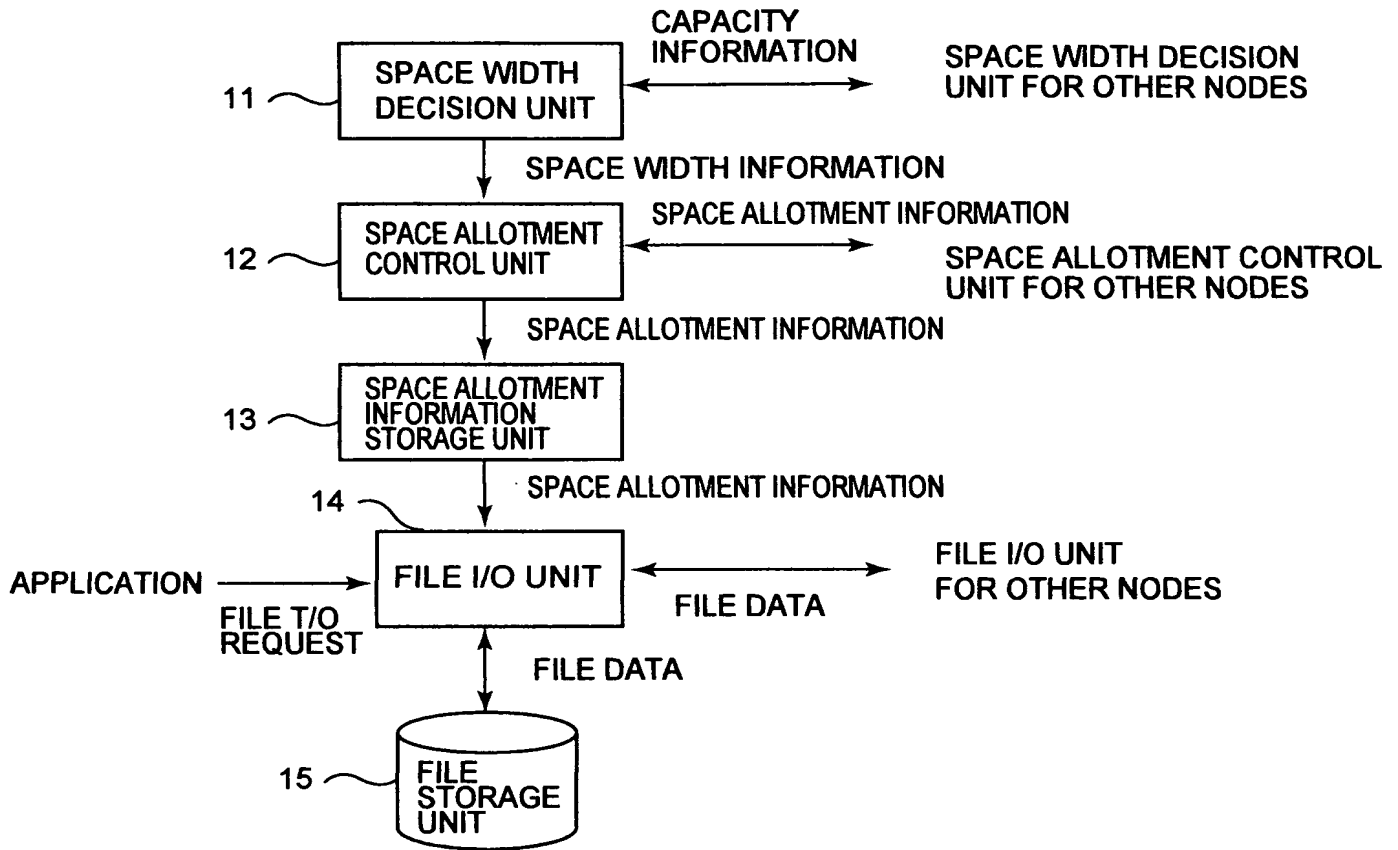CAPACITY BETWEEN TWO NEIGHBORING NODES)

FILE

~ 1

~ 1
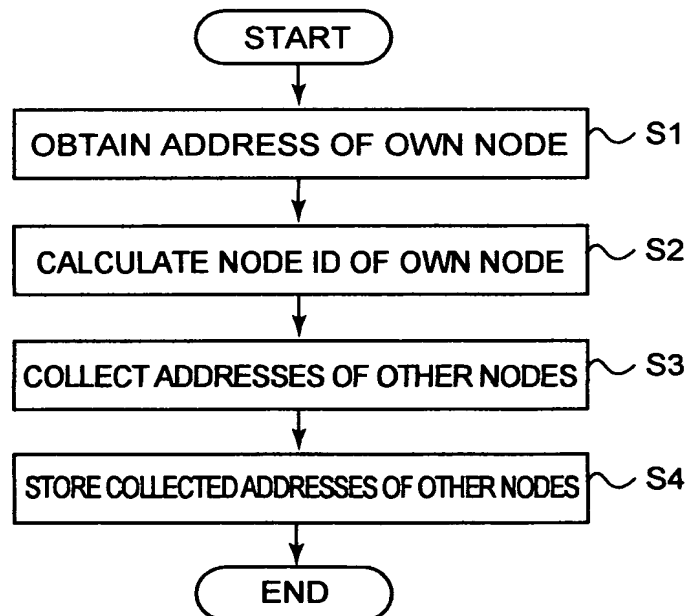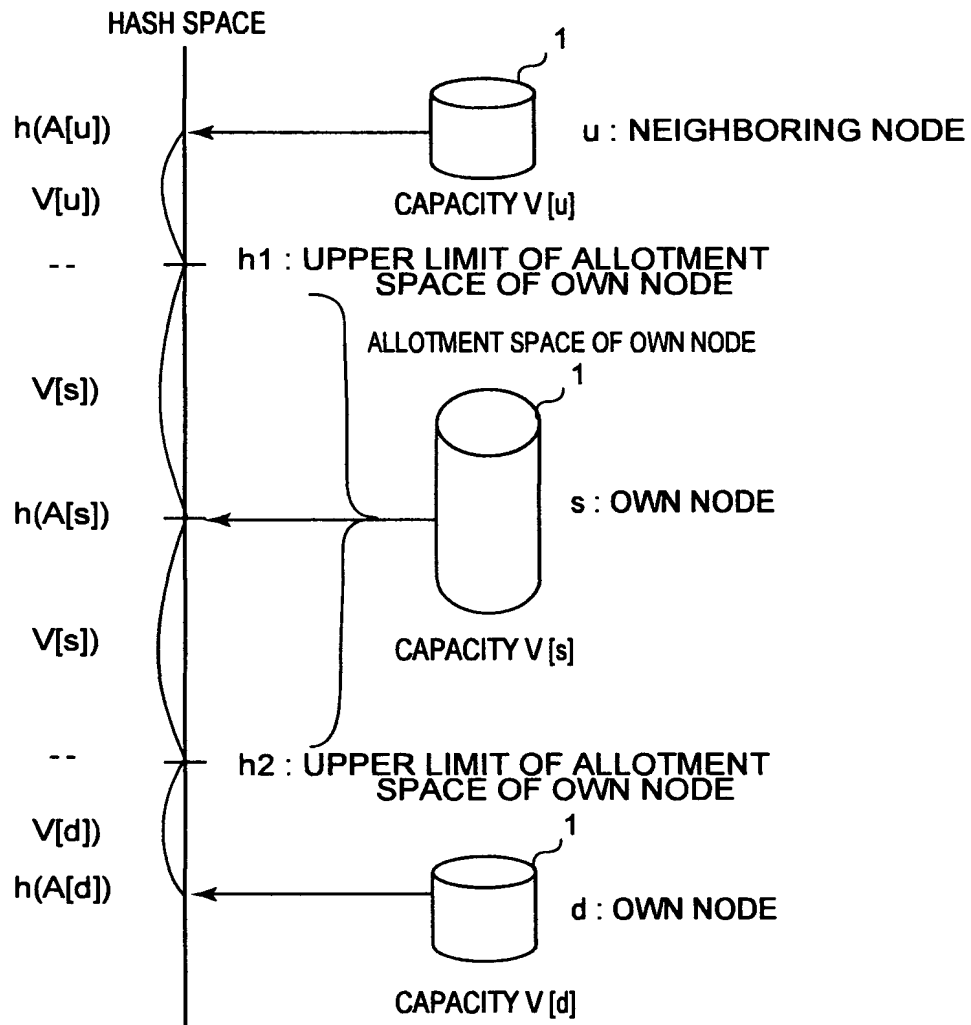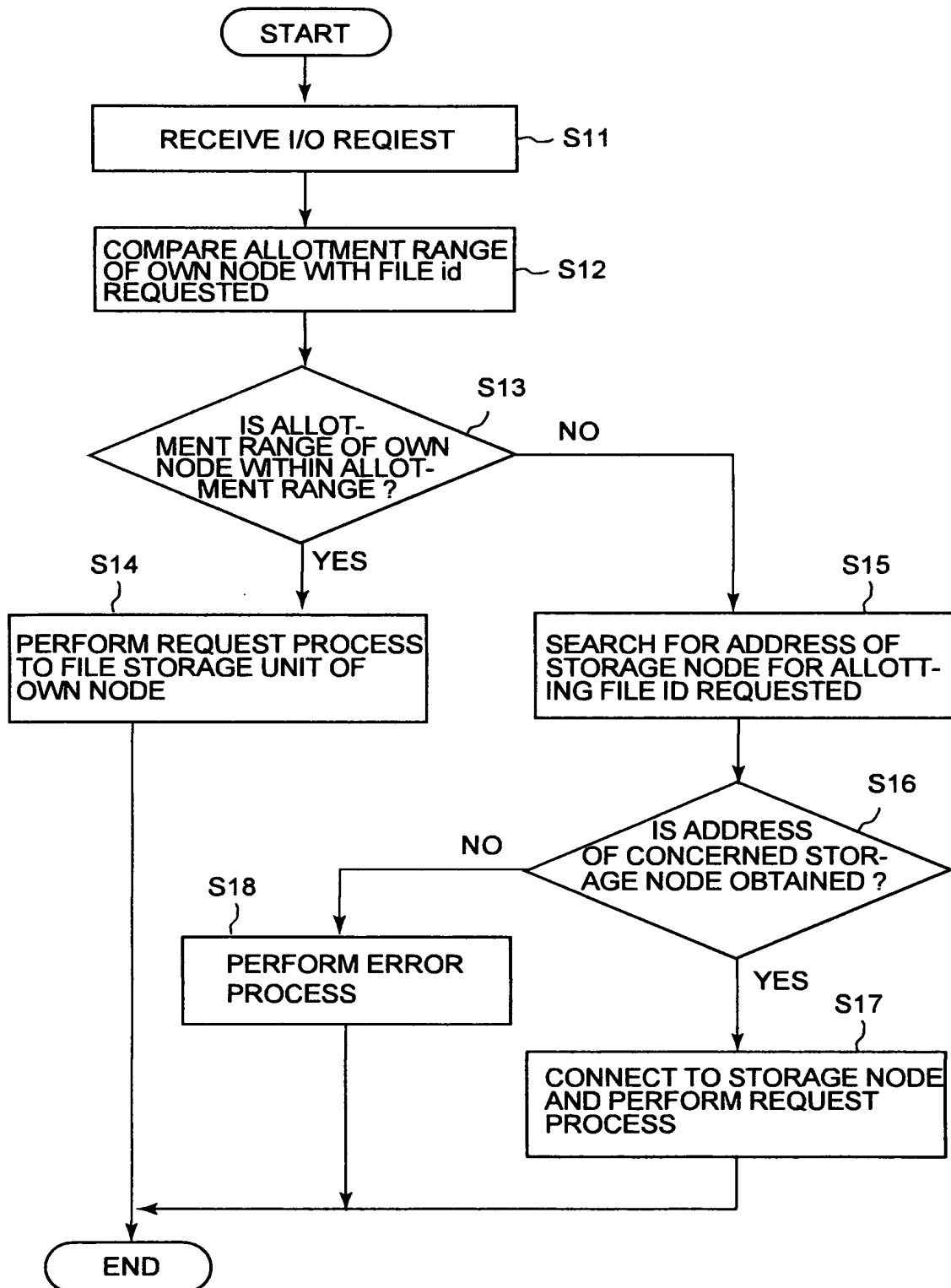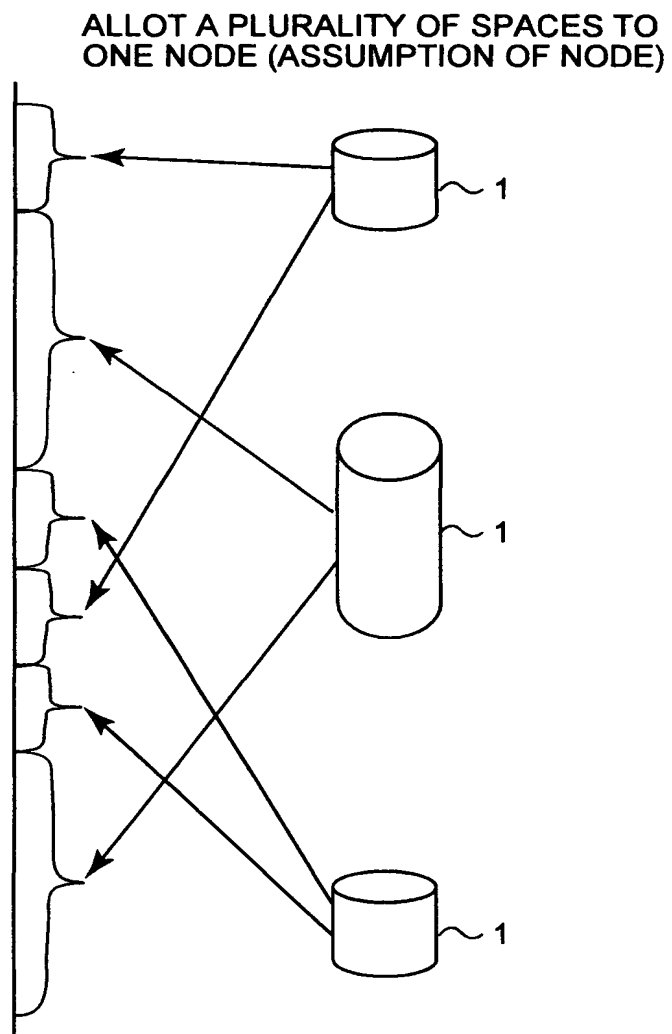
~ 1

~ 1

THE FILES ARE STORED IN
A PLURALITY OF NODES